DOI 10.29042/2018-3465-3469

Helix Vol. 8(4): 3465- 3469

Application of Interpolation Pooling in Convolutional Neural Networks

[1]Gaihua Wang, [*2]Guoliang Yuan, [3]Meng Lv, [4]WenZhou Liu

[1]Hubei Collaborative Innovation Centre for High-efficiency Utilization of Solar Energy, Hubei University of Technology, Wuhan 430068, China

[1,2,3,4] School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China

***Email**: guoliang_yuan@hotmail.com*

Received: 22nd March 2018, Accepted: 6th April 2018, Published:30th June 2018

Abstract In the existing convolutional neural networks, the majority of the used pooling operations are max pooling or mean pooling, but it would lose some important feature information when processing the feature maps. Here we report interpolation pooling to overcome the problem for retaining more effective information of feature maps. The interpolation pooling takes the known pixel points of 4x4 with the nearest to the interpolation point into account. Due to the distance from the pixels to be inserted, the weight of the pixels near the distance in the calculation is larger. We apply it to different convolutional neural networks, such as lenet-5 and pyramid convolutional neural networks. We found that the method has the advantages of faster convergence and higher accuracy than the traditional method of pooling.

Keywords: *Interpolation Pooling, Image Classification, Convolutional Networks*

1. Introduction

In recent years, deep learning has attracted the attention of many scholars. The fact has proved that deep learning has deeper and wider application than traditional shallow learning networks, including visual recognition, speech recognition and natural language processing. In 2006, Hinton[1] improved the method(deep belief nets) of deep learning breaks the bottleneck of the development of BP neural network. In all kinds of deep learning, convolutional neural networks (CNNs) have been the most extensively studied. CNNs consist of three types of layers: convolution, pooling and fully connected layer[2]. For convolutional layers, the convolution kernel is shared by all the spatial positions. which reduce the complexity of the model and make the network easier to train[3]. Pooling is an important concept of CNNs, including max pooling, mean pooling or mixed pooling. A pooling layer reduces computational load by reducing the number of convolutional layers. In 2012, Krizhevsky et al. proposed an AlexNet[4] model that shows significant improvements. AlexNet is similar to LeNet-5[3], but with a deeper structure. Simonyan et[5] proposed the VGG network based on AlexNet. And he proved that the enhancement of net work depth helps to improve the accuracy of image classification. By increasing the depth, the network can better approximate the objective function, increase the non-linearity, and get a better representation of the features. However, this also increases the complexity of the network and makes it more difficult to optimize. To solve degradation problem with increasing the depth of CNNs, He et[6] proposed a ResNet that won the 2015 ILSVRC championship. ResNet maps low-level features directly to high-level Network. And it is eight times as deep as VGG and 20 times faster than AlexNet. Szegedy et[7]. proposed an inception module by observing and optimizing the network structure, which reduces the network complexity and replaces the previous convolution kernel by using a 1 × 1 convolution kernel in the inception module. The number of training parameters for GoogLeNet[7] built using the Inception module is only 1 / 12th of AlexNet, but the accuracy of image classification on ImageNet is improved. In 2017, Saining Xie[8] proposed the ResNeXt network structure based on ResNet. ResNeXt improves the accuracy without increasing the complexity of the parameters and reducing the number of hyper-parameters. At the same time, a variety of methods [9-11]have been proposed to overcome the difficulties encountered in deep CNNs training.

All the methods mentioned above are improvements for the depth, activation function and convolution kernel of CNNs. In these models, max pooling or mean pooling is used. Max pooling simply selects the maximum value from the pooling area as the final response value, which is sensitive to noise information. Mean pooling takes the average values in the pooling area, which effectively reduces the impact of noise information, but it smooths the image and leads to the loss of high frequency information[12].In this paper, for the existing problems of the max pooling and the mean pooling in CNNs, interpolation pooling is proposed to optimize the network efficiency. Interpolation pooling mainly uses the method of image interpolation to select the nearest 16 pixels as the pixel value corresponding to the final image, so as to achieve the purpose of scaling the feature map.

2. CNNs

The basic structure of CNNs consists of an input layer, convolutional layer, pooling layer, fully connected layer. The convolutional layers and pooling layers are usually arranged alternately. The structure is as **Fig. 1**, one of which we used in our experiment.

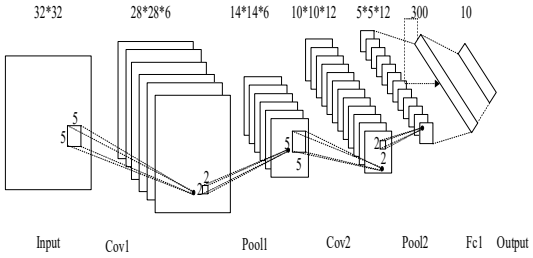

Fig. 2. CNNs Structure

The convolutional layer extracts different features of input through convolutional operation. The first layer of convolutional layer features such as edge, line and corner, while the higher layer convolution layer extracts more advanced features. The convolution form in CNNs[13] is as follows(Equation(1)).

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} k_{ij}^l + b_j^l\right) \tag{1}$$

Where l denote the current layer, x_j^l represents $j-th$ output of current layer, x_j^l represents input, M_j represents a selection of the input maps, k denotes the convolution kernel , b represents the bias added by each output feature map, and $f(\bullet)$ is the activation function with the sigmoid.

After the convolutional layer, the pooling layer is also composed of a plurality of feature maps, each of which uniquely corresponds to a feature maps of the upper layer, and does not change the number of feature maps. It just use the pooling operation function to further adjust the convolutional output. The maximum value of the pool to achieve the following methods(Equation(2))

$$S_j = \max_{i \in R_j}(a_i) \tag{2}$$

Where a_i represents the input feature map of the pooling layer, R_j represents the pooling area, and s_j is the pooling output feature map.

Mean pooling is implemented using the following methods(Equation(3)).

$$s_j = \frac{1}{|R_j|}\sum_{i \in R_j} a_i \tag{3}$$

3. The Proposed Method

We apply interpolation pooling to the pooling operation of convolutional neural networks. In order to facilitate the mutual comparison of different algorithms, we ensure that different pooling operation in the same kind of network. In this method, we need to interpolate the basis function to fit the data. The value of the function f at the point $p(x, y)$ can be obtained by a weighted average of the nearest sixteen sampling points in the rectangular grid, where we use two polynomial interpolation cubic functions. The following basis function is selected by us(Equation(4)).

$$S(w) = \begin{cases} 1-2|w|^2+|w|^2, |w|<1 \\ 4-8|w|+5|w|^2-|w|^2, 1 \le |w| < 2 \\ 0, |w| \ge 2 \end{cases} \tag{4}$$

The bicubic interpolation pooling is as follows(Equation(5))

$$f(i+u, j+v) = ABC \tag{5}$$

Where A,B and C all is a matrix ,and their form is as follows(Equation(6)(7)(8)).

$$A = [S(1+u) \quad S(u) \quad S(1-u) \quad S(2-u)] \tag{6}$$

$$B = \begin{bmatrix} f(i-1,j-2) & f(i,j-1) & f(i+1,j-2) & f(i+2,j-2) \\ f(i-1,j-1) & f(i,j-1) & f(i+1,j-1) & f(i+2,j-1) \\ f(i-1,j) & f(i,j) & f(i+1,j) & f(i+2,j) \\ f(i-1,j+1) & f(i,j+1) & f(i+1,j+1) & f(i+2,j+1) \end{bmatrix} \tag{7}$$

$$C = [S(1+v) \quad S(v) \quad S(1-v) \quad S(2-v)] \tag{8}$$

Where $f(i, j)$ represents the gray value of the pixel at(i, j). $f\left(i+u, j+v\right)$ indicates that the image to be calculated contains the pixel coordinates of the fractional part in the original image, u indicates the decimal coordinate in the horizontal direction, and v indicates the small coordinate in the vertical direction ,as shown **Fig. 2.**

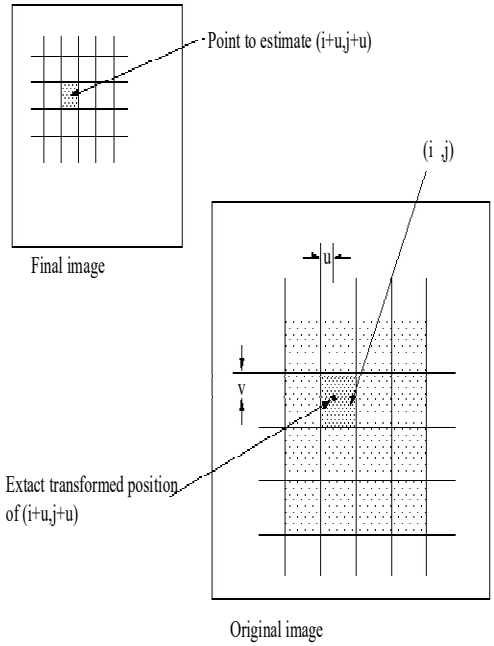

Fig. 2. Interpolation Pooling

Assuming the original image O pixel is m * m, the size of the final image F after the interpolation pool is $M * M$. According to the ratio we can determine the coordinates of the final image $F(X,Y)$ in O by Equation (9):

$$O(x,y)=O(X*\frac{M}{m},Y*\frac{M}{m})=(i+u,j+v) \quad (9)$$

According to the above formula to calculate the pixel value of all points of final image, we can write it as a function (Equation (10)):

$$F=\text{interpolation\_pool}(O,[M,M]) \quad (10)$$

Where O represents the input of the interpolation pool, F represents the output of the interpolation pool, and $[M,M]$ represents the size after pooling.

4. Experiments

We conduct experiments on three datasets: the MNIST standard dataset [3], the CIFAR-10[14, 15] dataset and the STL-10 dataset [16]. The results are compared with mean pooling and max pooling on three different convolutional network frameworks.

4.1 The MNIST Standard Dataset

We used different pooling methods with a CNN framework and lenet-5, respectively, their frameworks are same but pooling operation. The CNN and lenet-5 have two convolutional layers, two pooling layers, and two fully connected layers. The results are showed in the table1 and table 2.

Table 1. Network Structure: CNN

Test Accuracy (%)			
Iterations	Mean Pooling	Max Pooling	Our Method
10	97.04	97.57	98.04
20	98.00	98.09	98.40
150	98.43	98. 65	99.10

Table 2. Network Structure: Lenet-5

Test Accuracy (%)			
Iterations	Mean Pooling	Max Pooling	Our Method
10	97.44	97.81	98.21
20	98.55	98.66	98.90
150	98.94	98.96	99.15

It can be seen from the above experimental results that in the same convolutional neural network, different pooling operations will bring different effects. Average pooling classification is the worst, and the interpolation pooling is the best one. Test accuracy of our method is 98.04% when iterations is 10. Mean pooling is 97.04%, and max pooling is 97.57%. Further analysis can be found in both networks the advantage of interpolation pooling is that it increases the accuracy and accelerates the speed of the convolutional neural network, compared to the mean pooling and the max pooling.

4.2 CIFAR-10 Dataset

For the CIFAR-10 dataset, we use different pooling methods with a CNN network framework and pyramid CNNs[17]. Network structure of CNN is the same as above. pyramid CNNs has the same number of layers. The results are showed in the table 3 and table 4.

Table 3. Network Structure: CNNs

Test Accuracy (%)			
Iterations	Mean Pooling	Max Pooling	Our Method
10	32.16	37.41	41.87
50	47.88	54.30	54.54
100	52.06	56.99	57.61

Table 4. Network Structure: Pyramid CNNs

Test Accuracy (%)			
Iterations	Mean Pooling	Max Pooling	Our Method
10	42.84	40.12	51.95
20	48.44	50.08	56.23
70	60.16	61.21	64.75

Compared with the previous, when using the same network framework to test different datasets, the results vary greatly. We can found that when iterations is 10, Our method is 51.95% in test accuracy. but other is 42.84% and 40.12%. In terms of its pooling operation, interpolation pooling has better performance than mean pooling and max pooling.

4.3 STL-10 Dataset

The STL-10 dataset is an image recognition dataset used to develop unsupervised feature learning, deep learning, self-learning learning algorithms. It was inspired by the CIFAR-10 dataset, but with some modifications. For example, each of its classes has fewer training examples than CIFAR-10, but provides a large number of unlabeled samples. Due to its particularity, we chose 4 classes to conduct our tests. In order to test whether interpolation pooling has a universal existence, the network we use an unsupervised learning-based linear encoder through a pooling operation extracted features. The network structure we used can be found in UFLDL Tutorial[18]. We call it as unsupervised CNNs(UCNNs). it has just one convolutional layers, and one pooling layers.

Table 5. Network Structure: UCNNs

Test Accuracy (%)		
Mean Pooling	Max Pooling	Our Method
80.594	79.344	81.625
80.125	79.188	81.156
80.406	78.0563	81.506

It can be seen from the table 5 that the max pooling is relatively worst. Boureau et[19] mentioned that when analyzing the impact of different pooling operations, The max pooling operation and the mean pooling operation show different performances. And the experimental results conform to the description. The effect of interpolation pooling is still the best.

5. Conclusion and Future Work

In the framework of convolutional neural networks, this paper proposes interpolation pooling strategies that can be used in combination with other methods. The application of interpolation pooling can effectively prevent the occurrence of over-fitting and effectively improve the robustness of the model while retaining the high frequency information and effectively preventing the loss of information such as edges. In addition, interpolation pooling can be combined with any existing convolution neural network model. Experiments show that this method can effectively improve the convergence rate of the model while ensuring the accuracy of the model.

Acknowledgements

This work is supported by the National Nature Science Fund of China under Grant No. 61601176.

References

[1]. Hinton, G.E. and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks. Science, 2006. 313(5786): p. 504.

[2]. Gu, J., et al., Recent Advances in Convolutional Neural Networks. Computer Science, 2016.

[3]. LeCun, Y.L., et al., Gradient-based learning applied to document recognition. Proc IEEE. Proceedings of the IEEE, 1998. 86(11): p. 2278-2324.

[4]. Krizhevsky, A., I. Sutskever and G.E. Hinton. ImageNet classification with deep convolutional neural networks. in International Conference on Neural Information Processing Systems. 2012.

[5]. Simonyan, K. and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science, 2014.

[6]. He, K., et al., Deep Residual Learning for Image Recognition. 2015: p. 770-778.

[7]. Szegedy, C., et al. Going deeper with convolutions. in Computer Vision and Pattern Recognition. 2015.

[8]. Xie, S., et al., Aggregated Residual Transformations for Deep Neural Networks. 2016.

[9]. Lin, T.Y., A. Roychowdhury and S. Maji, Bilinear Convolutional Neural Networks for Fine-grained Visual Recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017. PP(99): p. 1-1.

[10]. Howard, A.G., et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017.

[11]. Badrinarayanan, V., A. Kendall and R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017. PP(99): p. 1-1.

[12]. Zeiler, M.D. and R. Fergus, Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. Eprint Arxiv, 2013.

[13]. Bouvrie, J., Notes on Convolutional Neural Networks. Neural Nets, 2006.

[14]. Krizhevsky, A., Learning Multiple Layers of Features from Tiny Images. 2009.

[15]. Hinton, G.E., et al., Improving neural networks by preventing co-adaptation of feature detectors. Computer Science, 2012. 3(4): p. págs. 212-223.

[16]. Coates, A., et al. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. in

  ISSN 2319 – 5592 (Online)

AISTATS. 2011.
[17]. Guanhao, W. and X. Jun, Fast feature representation method based on multi-level pyramid convolution neural network. Application Research of Computers, 2015. 32(8): p. 2492-2495.
[18]. Feature extraction using convolution - Ufldl. 2017.
[19]. Boureau, Y.L., J. Ponce and Y. Lecun. A Theoretical Analysis of Feature Pooling in Visual Recognition. in International Conference on Machine Learning. 2010.

